

CLAIM AMENDMENTS

Please amend claims as follows:

1. (Currently Amended) A method of garbage collecting in a storage device comprising:

locating blocks of data in a log that are both referenced by at least one other block of data ~~residing-stored~~ within a set of one or more storage trees[[,]] and within a range at a tail of the log using pruned walking, the range representing an address range within an allocated segment of the log,

~~wherein the log is implemented in a hierarchical architecture having a plurality of storage trees, each storage tree representing a snapshot taken at a point in time of target data being processed, each storage tree~~ representing active snapshots of user-generated data stored from each one of a plurality of remote computer systems taken at different points in time and having a plurality of nodes, and each node representing a block of data of the an active snapshot associated with each storage tree, the plurality of nodes including leaf nodes representing data blocks of the user-generated data stored from each one of the plurality of remote computer systems and root and intermediate nodes representing data blocks of system-generated data created by the storage device, the system-generated data including pointers to reference other data blocks of the user-generated data to enable overlapping and sharing of the user-generated data between and among different snapshots of the plurality of snapshots;

copying the blocks of data that are referenced by one or more other blocks of data of other nodes and within the range to an unallocated segment of the log, wherein blocks of data that are not referenced by other blocks of data and within the range remain untouched, and

marking the range at the tail of the log as unallocated so that at least a portion of an address space within the range can be reclaimed.

2. (Cancelled).

3. (Previously Presented) The method of claim 1, wherein locating the blocks of data that are referenced and within the range includes determining a minimum value among addresses of descendent nodes of a node, wherein the minimum value represents a minimum address offset of a node that is closest referenced from the blocks of data.

4. (Previously Presented) The method of claim 3, wherein a location table includes an entry for nodes that reference other nodes and wherein determining the minimum value among addresses of descendent nodes of the node includes retrieving the minimum value from an entry in the location table associated with the node.

5. (Original) The method of claim 4, wherein locating the blocks of data that are referenced and within the range includes processing the descendent nodes of

the node upon determining that the minimum value among the addresses of the descendent nodes is within the range.

6. (Original) The method of claim 5, comprising modifying the addresses of the copied blocks of data that are stored in the location table based on the new locations of the copied blocks of data in the log.

7. (Previously Presented) The method of claim 5, further comprising modifying the minimum value in the entry in the table associated with the node when the minimum value changes based on the new locations of the copied blocks of data that are associated with descendent nodes of the node.

8. (Currently Amended) A method comprising:
garbage collecting within a range of addresses in a storage system having a plurality of storage trees representing active snapshots of user-generated data stored from each one of a plurality of remote computer systems taken at different points in time, each storage tree having a plurality of nodes and having multiple references to a same block of data, wherein each block of data is stored within a log as contiguous data in a sequential order and wraps around to the beginning of the log once the end is reached so that the size of the log does not increase beyond its originally created size, the garbage collecting including:

pruning walking of the plurality of storage trees to determine active blocks of data within said range, where active blocks of data are those that

are referenced by at least one other block of data still stored within one of the plurality of storage trees, the pruning walking including:

determining, based on accessing in one of said plurality of storage trees a parent node that has a plurality of descendent nodes, that none of the plurality of descendant nodes are associated with blocks of data within the range; and

skipping the walking of the plurality of descendent nodes based on said determining,

wherein the active blocks determined to be in the range are copied out of the range and the range is marked as unallocated so that at least a portion of the address space within the range can be reclaimed.

9. (Original) The method of claim 8, wherein the blocks of data are stored in a log and the range is a segment of the log.

10. (Original) The method of claim 9, wherein the segment is at the tail of the log.

11. (Original) The method of claim 10, wherein the determining is performed by comparing a minimum offset of the plurality of descendent nodes against the range, wherein the minimum offset is accessed when walking the parent node and without walking the plurality of descendent nodes.

12. (Cancelled).

13. (Previously Presented) The method of claim 8, wherein the range is a segment at the tail of a log and said copying is from the said segment at the tail to a segment at the head of the log.

14. (Previously Presented) The method of claim 8, wherein said copying includes updating addresses of the copied blocks of data within a location table.

15. (Currently Amended) A method of garbage collecting in a storage system comprising:

performing the following operations until each block of data that is active in a range to be cleaned at a tail of a log of data is copied to a head of the log, wherein the range to be cleaned is a range of addresses in ~~a~~the storage system, the storage system having a plurality of storage trees representing active snapshots of user-generated data stored from each one of a plurality of remote computer systems taken at different points in time, each storage tree having a plurality of nodes, wherein ~~a~~each block of data is ~~associated with~~represented by a node of the storage tree, the operations including:

locating blocks of data in ~~a~~the log that are both referenced by at least one other block of data ~~residing stored~~within the nodes of the plurality of storage trees~~[[,]]~~ and within the range to be cleaned using pruned walking;

copying blocks of data associated with child nodes of a current node that are within the range to be cleaned to the head of the log;

retrieving a block of data associated with the current node, upon determining that a minimum address value among addresses of descendent nodes is within the range to be cleaned;

designating, as the current node, one of the child nodes of the current node that is an interior node, upon determining that at least one child node is an interior node;

designating, as the current node, an ancestor node of the current node whose descendent nodes are unprocessed; and

marking the range as unallocated when the blocks of data that are active and within the range are copied to the head of the log so that at least a portion of the address space within the range to be cleaned can be reclaimed.

16. (Original) The method of claim 15, wherein performing the following until each block of data that is active in the range to be cleaned at the tail of the log of data is copied to a head of the log includes updating addresses of that copied blocks of data within a location table.

17. (Original) The method of claim 15, wherein performing the following until each block of data that is active in the range to be cleaned at the tail of the log of data is copied to the head of the log includes updating a minimum address value among addresses of descendent nodes for an entry for the current node in a location table when the minimum address value changes based on copying of the blocks of data associated with the descendent nodes of the current node.

18. (Original) The method of claim 15, wherein at least one block of data stored in the log is referenced by more than one of other blocks of data.

19. (Cancelled).

20. (Currently Amended) A system comprising:

a storage device to store a number of blocks of data, wherein the blocks of data that are marked as allocated are non-modifiable, the blocks of data to be stored as a log, wherein each block of data within the log is stored as contiguous data in a sequential order and wraps around to the beginning of the log once the end is reached so that the size of the log does not increase beyond its originally created size; and

a garbage collection logic to locate the blocks of data that are both referenced by at least one other block ~~residing-stored~~ within a set of one or more storage trees, and within a range at a tail of the log using pruned walking, the range representing an address range to be cleaned within an allocated segment of the log,

wherein the log is implemented in a hierarchical architecture having a plurality of storage trees, each representing a snapshot taken at a point in time of target data being processed, each storage tree having a plurality of nodes, and each node representing a block of data of ~~the~~ an active snapshot associated with each storage tree,

and wherein said garbage collection logic is operable to copy the located blocks of data ~~that are referenced and within the range at the tail of the log to a~~ head of the log and mark the range to be cleaned as unallocated so that at least a portion of the address space within the range can be reclaimed.

21. (Original) The system of claim 20, wherein the garbage collection logic is to copy the blocks of data that are referenced to an unallocated address space of the log.

22. (Original) The system of claim 21, wherein the garbage collection logic is to copy the blocks of data that are referenced to a head of the log.

23. (Cancelled).

24. (Original) The system of claim 20, wherein at least one of the number of blocks of data are referenced by more than one reference.

25. (Original) The system of claim 20 comprising a location table to include entries associated with interior nodes of a storage tree, wherein each entry is to include a minimum value among the addresses of descendent nodes of the associated interior node.

26. (Original) The system of claim 25, wherein the garbage collection logic is to locate the blocks of data that are referenced and within the range at the tail of the log based on the minimum values stored in the entries of the location table.

27. (Currently Amended) A backup system comprising:

a plurality of storage trees, each storage tree representing an active snapshot of user-generated data stored from a file system located on a remote computer taken at a different time, each storage tree having a plurality of leaf nodes including leaf nodes representing data blocks of the user-generated data stored from the remote computer file system and root and intermediate nodes representing data blocks of system-generated data created by the storage device, the system generated data including pointers to reference other data blocks of the user-generated data to enable overlapping and sharing of the user-generated target data between and among different snapshots of the plurality of snapshots, wherein each leaf node ~~of said plurality of storage trees to include a block of~~ user-generated data from said remote computer file system that has been backed up from a set of one or more storage devices within the remote computer;

a storage space to store said blocks of data having been allocated from a backup storage space in said set of one or more storage devices;

a set of one or more location tables having stored therein a minimum address value for descendent nodes of interior nodes of said plurality of storage trees; and

a garbage collection logic to clean a currently selected range from the tail of said log, said garbage collection logic to perform pruned walking of nodes of said plurality of storage trees based on said set of location tables and said currently

selected range to locate blocks of data that are referenced by at least one other block of data currently ~~residing~~stored within the plurality of storage trees.

28. (Original) The backup system of claim 27, wherein two different nodes of a same storage tree reference a same node in the same storage tree.

29. (Original) The backup system of claim 27, wherein the garbage collection logic is to update references to a node that is within the currently selected range based on an update to an entry in the set of one or more location tables.

30. (Previously Presented) The backup system of claim 27, wherein the garbage collection logic is to prune walking of the nodes of said plurality of storage trees based on the minimum addresses stored in the set of one or more location tables.

31. (Currently Amended) An apparatus comprising:

a hardware backup system to backup a file system located on a remote computer, said backup file system including:

a tracking logic to generate a plurality of storage trees, each storage tree ~~having a plurality of nodes and representing~~ active backup snapshots of user-generated data stored from said remote computer file system taken at different times by recording references to blocks of backed up data stored in a set of one or more storage devices, each storage tree having a plurality of nodes including leaf nodes representing data blocks of the user-generated data stored from the remote computer file system and root and intermediate nodes representing data blocks of

system-generated data created by the storage device, the system-generated data including pointers to reference other data blocks of the user-generated data to enable overlapping and sharing of the user-generated target data between and among different snapshots of the plurality of snapshots;

an allocator logic to allocate contiguous blocks of storage space from a log of a back up storage space to store said blocks of backed up data;

a garbage collection logic to, responsive to deletion of one or more of said active backup snapshots, clean a currently selected contiguous range from the tail of said log, said garbage collection logic to,

walk only those nodes of said plurality of storage trees that possibly identify those of said blocks of data that are stored in said currently selected contiguous range or that possibly are themselves stored in said currently selected contiguous range,

locate blocks of data that are referenced by at least one other block of data ~~residing stored~~ within the plurality of storage trees, and

sweep said currently selected contiguous range, copying blocks of data that are referenced and within the range out of the range and marking the range as unallocated so that at least a portion of the address space within the range can be reclaimed.

32. (Currently Amended) The apparatus of claim 31, wherein the plurality of storage trees include interior nodes and leaf nodes, the interior nodes to include references to other nodes in one or more of the plurality of storage trees, wherein

two different interior nodes of a same tree references a same node in the same tree.

33. (Currently Amended) A ~~machine~~computer-storage medium that stores instructions, which when executed by a ~~machine~~computer, cause said ~~machine~~computer to perform garbage collection operations in a hardware storage device comprising:

locating blocks of data in a log that are both referenced by at least one other block of data ~~residing~~stored within a plurality of storage trees~~[[,]]~~ and within a range at a tail of the log using pruned walking, the range representing an address range to be cleaned within an allocated segment of the log, ~~wherein the log is implemented in a hierarchical architecture having the plurality of storage trees, each storage tree representing a snapshot taken at a point in time of target data being processed, each storage tree~~ representing active snapshots of user-generated data stored from each one of a plurality of remote computer systems at different points in time and having a plurality of nodes, and each node representing a block of data of an active snapshot associated with each storage tree, wherein each block of data within the log is stored as contiguous data in a sequential order and wraps around to the beginning of the log once the end is reached so that the size of the log does not increase beyond its originally created size;

copying the blocks of data that are referenced and within the range to an unallocated segment of the log, wherein blocks of data that are not referenced by other blocks of data and within the range remain untouched, and

marking the range at the tail of the log as unallocated so that at least a portion of an address space within the range can be reclaimed.

34. (Cancelled).

35. (Currently Amended) The ~~machine~~-computer-storage medium of claim 33, wherein locating the blocks of data that are referenced and within the range includes determining a minimum value among addresses of descendent nodes of a node.

36. (Currently Amended) The ~~machine~~-computer-storage medium of claim 35, wherein a location table includes an entry for nodes that reference other nodes and wherein determining the minimum value among addresses of descendent nodes of the node includes retrieving the minimum value from an entry in the location table associated with the node.

37. (Currently Amended) The ~~machine~~-computer-storage medium of claim 36, wherein locating the blocks of data that are referenced and within the range includes processing the descendent nodes of the node upon determining that the minimum value among the addresses of the descendent nodes is within the range.

38. (Currently Amended) The ~~machine~~-computer-storage medium of claim 37 comprising modifying the addresses of the copied blocks of data that are stored in the location table based on the new locations of the copied blocks of data in the log.

39. (Currently Amended) The ~~machine-computer~~-storage medium of claim 37 comprising modifying the minimum value in the entry in the table associated with the node when the minimum value changes based the new locations of the copied blocks of data that are associated with descendent nodes of the node.

40. (Currently Amended) A ~~machine-computer~~-storage-medium that stores instructions, which when executed by a ~~machine-computer~~, cause said ~~machine computer~~ to perform operations in a hardware storage device comprising:
~~perform garbage collection operations in a hardware storage device comprising~~
garbage collecting within a range of addresses in ~~a~~-the storage system having a plurality of storage trees, each storage tree representing active snapshots of user-generated data stored from each one of a plurality of remote computer systems taken at different times and having a plurality of nodes, the plurality of nodes including leaf nodes representing data blocks of the user-generated data stored from each one of the plurality of remote computer systems and root and intermediate nodes representing data blocks of system-generated data created by the storage device, the system-generated data including pointers to reference other data blocks of the user-generated data to enable overlapping and sharing of the user-generated target data between and among different snapshots of the plurality of snapshots and having multiple references to the same block of data, the garbage collecting including:

pruning walking of the plurality of storage trees to determine active blocks of data within said range, where active blocks of data are those that are referenced by at least one other block of data still in one of the plurality of storage trees, the pruning walking including:

determining, based on accessing in one of said plurality of storage trees a parent node that has a plurality of descendent nodes, that none of the plurality of descendant nodes are associated with blocks of data within the range; and

skipping the walking of the plurality of descendent nodes based on said determining,

wherein the active blocks determined to be in the range are copied out of the range and the range is marked as unallocated so that at least a portion of the address space within the range can be reclaimed.

41. (Currently Amended) The ~~machine-computer~~-storage medium of claim 40, wherein the blocks of data are stored in a log and the range is a segment of the log.

42. (Currently Amended) The ~~machine-computer~~-storage medium of claim 41, wherein the segment is at the tail of the log.

43. (Currently Amended) The ~~machine-computer~~-storage medium of claim 42, wherein the determining is performed by comparing a minimum offset of the plurality of descendent nodes against the range, wherein the minimum offset is

accessed when walking the parent node and without walking the plurality of descendent nodes.

44. (Cancelled).

45. (Currently Amended) The ~~machine-computer~~-storage medium of claim 43, wherein the range is a segment at the tail of a log and said copying is from the said segment at the tail to a segment at the head of the log.

46. (Currently Amended) The ~~machine-computer~~-storage medium of claim 43, wherein said copying includes updating addresses of the copied blocks of data within a location table.

47. (Currently Amended) A ~~machine-computer~~-storage medium that stores instructions, which when executed by a ~~machine-computer~~, cause said ~~machine computer~~ to perform garbage collection operations in a hardware storage device comprising:

performing the following operations until each block of data that is active in a range to be cleaned at a tail of a log of data is copied to a head of the log, wherein the range to be cleaned is a range of addresses in a storage system having a plurality of storage trees, each storage tree having a plurality of nodes, wherein a block of data is associated with a node of the storage tree, and wherein data stored within the log is stored as contiguous data in a sequential order and wraps around

to the beginning of the log once the end is reached so that the size of the log does not increase beyond its originally created size, the operations including:

locating blocks of data in a log that are both referenced by at least one other block of data ~~residing~~stored within the plurality of storage trees, and within the range using pruned walking;

copying blocks of data associated with child nodes of a current node that are within the range to be cleaned to the head of the log;

retrieving a block of data associated with the current node, upon determining that a minimum address value among addresses of descendent nodes is within the range to be cleaned;

designating, as the current node, one of the child nodes of the current node that is an interior node, upon determining that at least one child node is an interior node;

designating, as the current node, an ancestor node of the current node whose descendent nodes are unprocessed; and

marking the range at the tail of the log as unallocated when the blocks of data that are active and within the range are copied to the head of the log so that at least a portion of an address space within the range can be reclaimed.

48. (Currently Amended) The ~~machine-computer~~storage medium of claim 47, wherein performing the following until each block of data that is active in the range to be cleaned at the tail of the log of data is copied to a head of the contiguous log includes updating addresses of that copied blocks of data within a location table.

49. (Currently Amended) The ~~machine-computer~~-storage medium of claim 47, wherein performing the following until each block of data that is active in the segment to be cleaned at the tail of a log of data is copied to the head of the log includes updating a minimum address value among addresses of descendent nodes for an entry for the current node in a location table when the minimum address value changes based on copying of the blocks of data associated with the descendent nodes of the current node.

50. (Currently Amended) The ~~machine-computer~~-storage medium of claim 47, wherein at least one block of data stored in the log is referenced by more than one of other blocks of data.

51. (Cancelled).